smiths aerospace
bringing technology to life

Development, Validation, and Demonstration of Technologies for HUMS Automated Testing

Presentation by: Andy Brooks

FAA Semi-annual Rotorcraft Structures R&D Review.

13 February 2007
William J. Hughes Technical Center, Atlantic City, NJ

www.smiths-aerospace.com

Presentation Topics

Research Objectives

Review of 2006 Research

- Industry Techniques and Best Practices
- Tools Complementing Best Practices
- Benefits to HUMS
- System Functionality Partition
- Tool and Process Recommendations
- Cost/Benefit Analysis and Recommendations

Plan for 2007 Research

- Research Areas
- Demonstration Project
- Artifacts
- 2007 Schedule

Program Status

Research Conclusions

Research Objectives

Objectives

- Research and define current state-of-the-art tools, techniques, and best practices that can be applied to an HGS Automated Test Environment.
- Demonstrate that implementation of automated testing for an HGS fulfills the criteria for certification and subsequent credit validation requirements defined in AC-29-2C, Section MG-15.

Task Summary

Task 1: Detailed Work Plan	Year 1
Task 2: Reports	Ongoing
Task 3: Research Tools, Techniques and Best Practices	Year 1
Task 4: Research Effect of ATE on Certification Process	Year 2
Task 5: Implementation and Technology Transfer Plan	Year 2

2006 Research

Review of 2006 Research

- Industry Techniques and Best Practices
- Tools Complementing Best Practices
- Benefits to HUMS
- System Functionality Partition
- Tool and Process Recommendations
- Cost/Benefit Analysis and Recommendations

Industry Techniques and Best Practices

- Test Driven Development
- Short Iterations
- Continuous Integration

Tools Complementing Best Practices

Directly

- Continuous Integration
- Unit Test Framework
- One-Step Build
- Mock Tools

Indirectly

- Developmental Configuration Management
- Documentation
- Requirements Management / Project Management
- Static Analysis
- Dynamic Analysis

Continuous Integration

A Continuous Integration tool automates the integration and test cycle during development.

- Initiates ‘builds’ when integration is required
  - Initiates builds based on changes detected in DCM.
  - Initiates timed (e.g. nightly) builds.
- The build typically compiles and tests the software.
- Publishes build results and build artifacts.

Examples of Continuous Integration tools

CruiseControl, Mozilla Tinderbox, AnthillPro, DamageControl, IBM BuildForge, BuildBot, Apache Continuum

Preliminary Assessment by Smiths DERs

Does not automate, eliminate or reduce any DO-178B process.

Unit Test Framework

The unit test framework is used to aid developers in writing programmatic automated tests.

Typically a simple software library written in the native language for the application being tested.

Usually contains functionality for asserting failures, exception expectations, reporting test results.

Examples of Unit Test Frameworks

JUnit (Java), CPPUnit (C++), NUnit (.NET)

Preliminary Assessment by Smiths DERs

Would require qualification as a Software Verification Tool

One-Step Build

Tool that allows source code and tests to be compiled and run consistently without direct user interaction.

Combines functions of scripting language with functions of a typical ‘make’ program.

Often extendable to perform custom tasks required by a particular build.

Examples of One-Step Build tools

Apache Ant, Maven, Rake, Make, SCons

Preliminary Assessment by Smiths DERs

Does not automate, eliminate or reduce any DO-178B process.

Mock Tools

Tools for producing test software that mocks (simulates) the functionality of the parts of the system not under test. Typically this is a simple software library written in the native language for the application being tested.

Can be used to:

- Isolate tests from errors outside the functionality under test.
- Supply undeterminable results (accelerometer output) or hard to produce error conditions.
- Test functions where related functionality does not yet exist.
- Improve speed of test execution.

Examples of Mock Tools

NMock (.NET), jMock (Java), EasyMock (Java), MockPP (C++)

Preliminary Assessment by Smiths DERs

Would require qualification as a Software Verification Tool

Benefits to HUMS

- Time and Cost to initial certification could be reduced.
- In general, problems with fielded systems would more likely be addressed.
- In general, problems with fielded systems would be addressed earlier.
- In particular, problems related to performance (e.g. Speed, Reliability) with fielded systems would more likely be addressed.
- Increasing the scope of HUMS by interfacing to other Aircraft systems would become an easier task.
- Could allow HUMS to move to a higher degree of certification, and increase its effectiveness.

System Functionality Partition

Automated Testing and System Functionality Partitioning complement each other and can both be used to reduce the time and cost of certifying a HUMS.

Methods of partitioning

- Partitioning by Criticality
- Partitioning by Testability
- Partitioning by Probability of Change

Introducing automated testing had no effect on how the partitioning would be validated.

Tool and Process Recommendations

Tool Recommendations

- Use tools that integrate well with your existing environment
  - Programming Language.
  - Version Control.
  - Operating System.
- Good starting points
  - Continuous Integration – CruiseControl
  - Build Automation – Apache Ant
  - Test Framework – xUnit

Process Recommendations

- Automated Testing standards
- Build Rules
- Test Driven Development

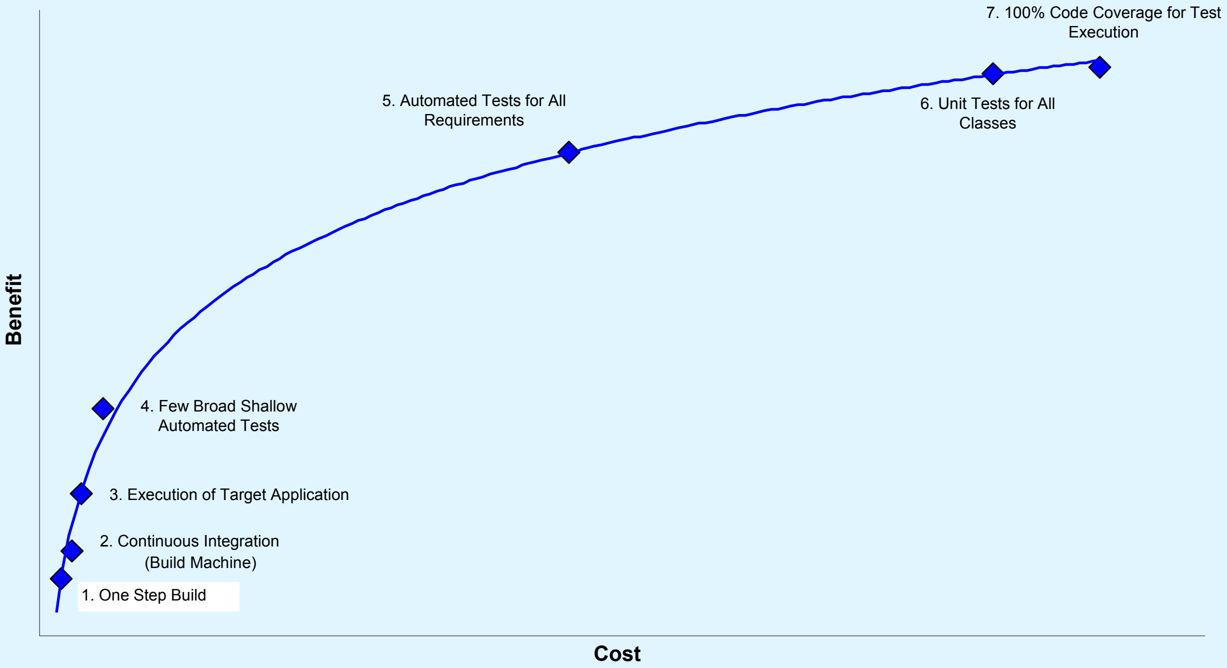
Cost/Benefit Analysis and Recommendations

The table shows successive stages of implementation of a completely automated and tested build in the typical order that a project would implement them. For the most part later steps build upon the earlier steps.

Technology Level	Estimated Cost	Cumulative % Cost	Cumulative Benefit
1. One Step Build	100 h	2%	10%
2. Continuous Integration (Build Machine)	50 h	3%	15%
3. Execution of Target Application	50 h	4%	25%
4. Few Broad Shallow Automated Tests	100 h	6%	40%
5. Automated Tests for All Requirements	2200 h	50%	85%
6. Unit Tests for All Classes	2000 h	90%	99%
7. 100% Code Coverage for Test Execution	500 h	100%	100%

Cost/Benefit Analysis and Recommendations

The graph shows that after Level 4 (Few Broad Shallow Automated Tests) the returns start to diminish, and for legacy projects the decision on how far to go will probably depend on the life of the project. It is expected that it will always be a good investment to get to Level 4.



Plan for 2007 Research

Plan for 2007 Research

- Research Areas
- Demonstration Project
- Artifacts
- 2007 Schedule

Research Areas

- Certification Issues for Automated Testing
- Demonstration Project
- Automated Testing Tool Qualification
- Effect on planning and processes.

Demonstration Project

This is a demonstration, showing typical HUMS functionality.

Allows certification to be demonstrated with the focus on the automated testing rather than peripheral project specific issues.

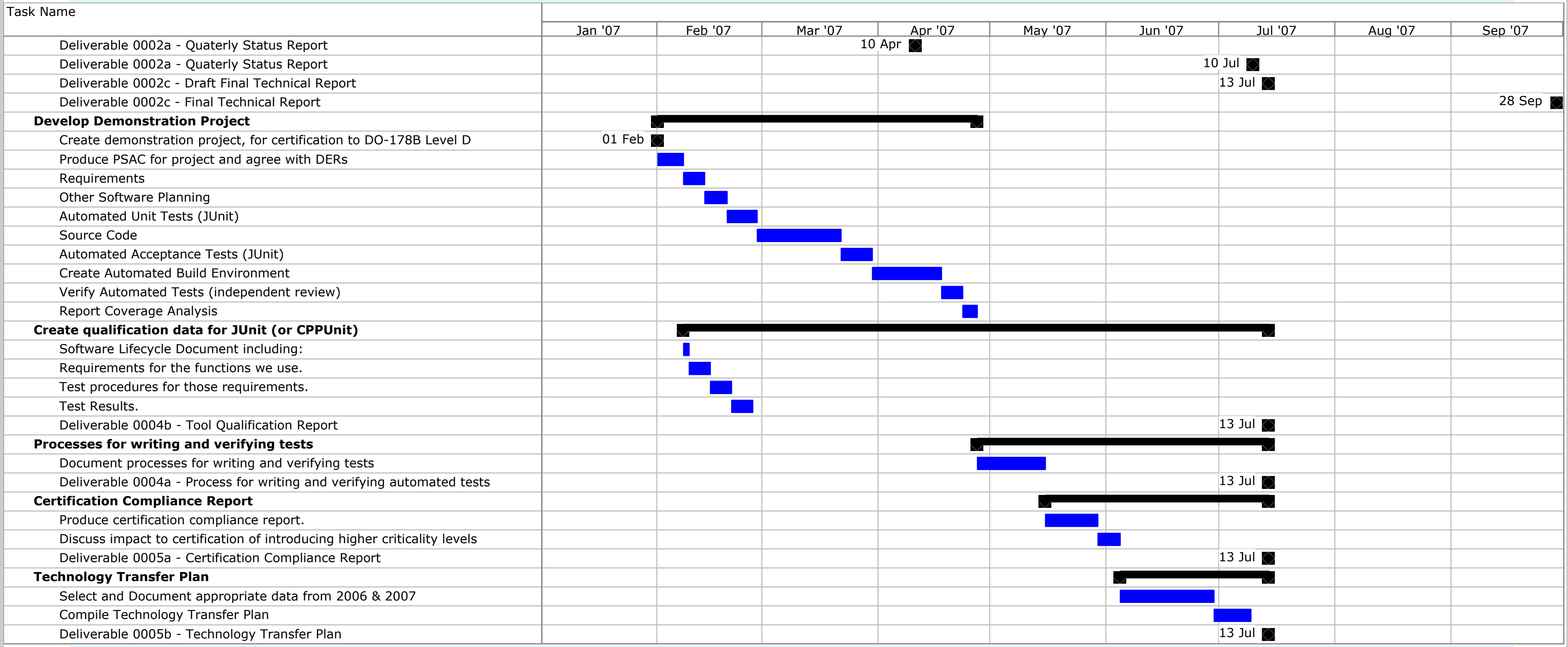
Selection Criteria

- Simple Source Code
- Plenty of obvious test cases
- Plenty of error conditions
- Simple IO / No GUI.

Artifacts

- Certification Compliance Report
- Processes related to writing and verifying automated tests.
- Qualification data for an automated testing tool.
- All demonstration project artifacts.
- Technology Transition Plan

2007 Schedule



Task Name	
Deliverable 0002a - Quaterly Status Report	10 Apr
Deliverable 0002a - Quaterly Status Report	10 Jul
Deliverable 0002c - Draft Final Technical Report	13 Jul
Deliverable 0002c - Final Technical Report	28 Sep
Develop Demonstration Project	
Create demonstration project, for certification to DO-178B Level D	01 Feb
Produce PSAC for project and agree with DERs	
Requirements	
Other Software Planning	
Automated Unit Tests (JUnit)	
Source Code	
Automated Acceptance Tests (JUnit)	
Create Automated Build Environment	
Verify Automated Tests (independent review)	
Report Coverage Analysis	
Create qualification data for JUnit (or CPPUnit)	
Software Lifecycle Document including:	
Requirements for the functions we use.	
Test procedures for those requirements.	
Test Results.	
Deliverable 0004b - Tool Qualification Report	13 Jul
Processes for writing and verifying tests	
Document processes for writing and verifying tests	
Deliverable 0004a - Process for writing and verifying automated tests	13 Jul
Certification Compliance Report	
Produce certification compliance report.	
Discuss impact to certification of introducing higher criticality levels	
Deliverable 0005a - Certification Compliance Report	13 Jul
Technology Transfer Plan	
Select and Document appropriate data from 2006 & 2007	
Compile Technology Transfer Plan	
Deliverable 0005b - Technology Transfer Plan	13 Jul

Program Status

Budget Expenditure

Expenditure at December 2006, 16% of total award.

Issues/Concerns

Validation of the System Functionality Partition.

Accomplishments

Completed Task 3: Research Tools, Techniques and Best Practices.

Deliverables

- Report 0003a - Assessment of Tools, Techniques and Best Practices Available to Assist the Automated Testing of HUMS Software.
- Report 0003b - Recommendations, Benefits and Examples of Automated Testing in relation to HUMS Software.
- Annual Technical Report.

Research Conclusions

The 2006 research has determined that:

- The tools and technologies being researched are appropriate for HUMS.
- There are benefits for both legacy and green-field HUMS.
- A small subset of the tools and technologies offer the most benefit for cost.

The 2007 research should determine that:

- The assumed certification benefits are valid and feasible.
- Other HUMS developers can easily exploit this technology following our research.

Questions?